Penggunaan Algoritma RSA dengan Metode The Sieve of Eratosthenes dalam Enkripsi dan Deskripsi Pengiriman Email

Muhammad Safri Lubis
Jurusan Teknologi Informasi
Fak. Ilmu Komputer dan Teknologi Informasi, USU
Medan, Indonesia
safri@usu.ac.id

Mohammad Andri Budiman
Jurusan Ilmu Komputer
Fakultas Ilmu Komputer dan Teknologi Informasi, USU
Medan, Indonesia
mandrib@gmail.com

Karina Lolo Manik
Jurusan Teknologi Informasi
Fakultas Ilmu Komputer dan Teknologi Informasi, USU
Medan, Indonesia
qharin.id@gmail.com

***Abstrak*—Dewasa ini sistem informasi semakin berkembang. Hal ini dapat dilihat dari kemajuan teknologi dalam proses komunikasi yang dapat dilakukan secara jarak jauh dari dua tempat yang berbeda. Perkembangan teknologi telekomunikasi dengan menggunakan komputer menyebabkan mobilitas masyarakat semakin baik karena komunikasi bisa dilakukan tanpa perlu interaksi langsung satu sama lain. Berbagai jenis layanan komunikasi tersedia di internet seperti pengiriman pesan melalui *email* yang semakin diminati oleh masyarakat. Meningkatnya pemanfaatan layanan e-mail melalui internet menyebabkanpermasalahan juga bermunculan selain permasalahan adanya *hacker* dan *cracker*. Hal tersebut sangat memungkinkan pesan yang dikirim dapat disadap dan diubah oleh pihak lain. Salah satu bentuk pencegahan kejahatan di internet adalah dengan menggunakan algoritma kunci publik RSA, yaitu dengan cara mengenkripsi pesan yang akan dikirim menggunakan kunci publik yang telah dibangkitkan oleh pihak pengirim. Cara ini akan memperkuat tingkat keamanan pesan yang dikirim dalam suatu jaringan internet selama kunci privat terjaga kerahasiaannya. Oleh karena itu, algoritma RSA akan diterapkan pada suatu perangkat lunak yang dirancang dan dibuat mengunakan bahasa pemrograman *Visual Basic .NET* 2008 sebagai sarana untuk meningkatkan keamanan pengiriman pesan.**

Kata kunci—Internet; email; algoritma; RSA; kunci public; kunci privat;

I. PENDAHULUAN

Saat ini internet sering digunakan untuk mengirim pesan dengan menggunakan fasilitas yang disebut *email*karena sangat efisien, cepat, dan murah. Namun ada beberapa ancaman saat menggunakan *email* seperti penyadapan isi *email*, merubah isi *email* dan menjadikan *email* itu tidak asli lagi. Otentikasi adalah konsep yang dipakai untuk menjaga pesan yang dikirim agar tetap utuh dan asli (Firasyan,2011). Konsep otentikasi ini merupakan salah satu aspek yang diberikan dalam bidang kriptografi yang mempelajari tentang teknik-teknik menyandikan sebuah informasi dengan memanfaatkan model-model algoritma. Salah satu algoritma yang dapat digunakan dalam kegiatan enkripsi dan deksripsi pengiriman email adalah dengan menggunakan RSA. RSA merupakan algoritma asimetrik yang mempunyai dua kunci berbeda, yaitu kunci publik (untuk enkripsi) dan kunci privat (untuk dekripsi). Kunci-kunci yang ada pada pasangan kunci mempunyai hubungan secara matematis, tetapi tidak dapat dilihat secara komputasi untu mendeduksi kunci yang satu ke pasangannya. Dengan kelebihan ini diharapkan algoritma RSA dapat membantu keamanan dalam proses pengiriman sebuah email. Tulisan ini merupakan pembahasan dari pemanfaatan alogritma RSA dengan menggunakan bahasa pogram VB.Net sebagai sarana untuk meningkatkan keamananan pengiriman pesan.

II. METODOLOGI

A. *Bilangan Prima dengan Metode The Sieve Of Eratosthenes*

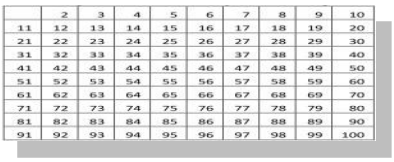
The Sieve Of Eratosthenes merupakan sebuah algoritma klasik untuk menentukan seluruh bilangan prima sampai bilangan N yang ditentukan. Cara kerja dari metode ini adalah dengan melakukan eliminasi bilangan yang bukan bilangan prima untuk menyaring suatu kumpulan bilangan menjadi kumpulan bilangan prima (Alghazali, 2010). Langkah-langkah penggunaannya sebagai berikut:

1. Tuliskan daftar bilangan dari 2 sampai batas atas bilangan yang akan dicari.
2. Tandai bilangan di dalam daftar yang merupakan kelipatan 2, dengan membiarkan bilangan 2 tetap tidak ditandai.

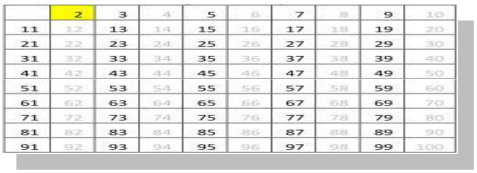
3. Lanjutkan ke bilangan berikutnya (dalam tahap ini adalah bilangan 3 dan seterusnya), dan tandai setiap kelipatan 3 dan seterusnya, dengan tetap membiarkan bilangan 3 dan bilangan selanjutnya tidak ditandai.
4. Lanjutkan langkah penandaan seperti di atas sampai batas atas bilangan yang ditentukan.

Contoh di bawah ini adalah untuk menentukan deretan bilangan prima dari 2 sampai 100 dapat dilakukan dengan langkah-langkah berikut ini:

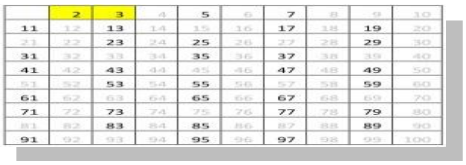
1. Buat daftar bilangan dari 2 sampai 100.



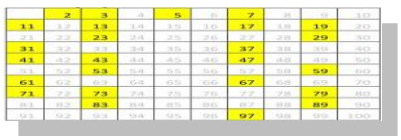
2. Bilangan terkecil yang tidak ditandai adalah bilangan prima, yaitu 2.
3. Beri tanda atau coret semua kelipatan bilangan 2 pada daftar bilangan yang ada.



4. Bilangan 3 merupakan bilangan prima berikutnya. Beri tanda semua kelipatan bilangan 3.



5. Ulangi langkah di atas untuk bilangan prima selanjutnya. Karena bilangan 4 sudah ditandai, lanjut ke bilangan 5. Beri tanda setiap kelipatan bilangan 5.
6. Berikutnya beri tanda bilangan 7 karena bilangan 6 sudah ditandai.
7. Bilangan selanjutnya adalah 11 karena bilangan 8, 9, dan 10 sudah ditandai. Namun, bilangan 11 sudah melewati 10 yang merupakan √100. Oleh karena itu, perulangan dihentikan.
8. Semua bilangan yang belum ditandai adalah bilangan prima.



9. Jadi deretan bilangan prima yang dihasilkan antara lain 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89 dan 97.
10. Bilangan di atas akan digunakan pada proses pembangkitan kunci yang akan dibahas pada bagian berikutnya.

B. Algoritma Kriptografi RSA

Algoritma kriptografi RSA didesain sesuai fungsinya sehingga kunci yang digunakan untuk enkripsi berbeda dari kunci yang digunakanuntuk dekripsi. Algoritma RSA disebut kunci publik karena kunci enkripsi dapat dibuat publik yang berarti semua orang boleh mengetahuinya, namun hanya orang tertentu yang dapat melakukan dekripsi terhadap pesan tersebut. Keamanan algoritmaRSA didasarkan pada sulitnya memfaktorkan bilangan besar menjadi faktor-faktor primanya (Sulistyanto, 2004). Besaran-besaran yang digunakan pada algoritma RSA adalah:

1. p dan q merupakan bilangan prima yang diambil secara acak. Bilangan prima tersebut dipilih langsung oleh pihak yang akan menerima pesan. Sifat dari kedua bilangan ini adalah rahasia. Ini berarti hanya pihak pengirim dan penerima saja yang mengetahuinya. Semakin besar p dan qmaka semakin baik (aman).
2. n = p.q, sifat dari n tidak rahasia, artinya orang lain dapat mengetahuinya.
3. $\phi(n) = (p-1)(q-1)$, sifat bilangan ini adalah rahasia.
4. e (kunci enkripsi), sifatnya tidak rahasia.
5. d (kunci dekripsi), sifatnya rahasia.
6. P (plainteks), merupakan informasi awal yang bersifat rahasia.
7. C (cipherteks), merupakan informasi yang telah dienkripsi dan bersifat tidak rahasia.

C. Konsep Dasar Perhitungan Matematis

Dalam setiap proses pada algoritma RSA terdapat perhitungan matematis. Pada proses pembangkitan kunci dibutuhkan perhitungan untuk menentukan nilai *Totient* n dan perhitungan dengan algoritma *Euclidean* untuk menentukan nilai dua buah bilangan yang relatif prima. Sedangkan padaproses enkripsi dan dekripsi dilakukan perhitungan menggunakan metode *Fast Exponentiation*.

1). Fungsi Totient Euler φ: Fungsi *Totient eulerφ* atau biasa disebut dengan fungsi *euler* merupakan salah satu fungsi yang dipakai dalam perhitungan matematis pada algoritma RSA. Fungsi *euler* mendefinisikan φ(n) untuk $n \geq 1$ yang menyatakan jumlah bilangan bulat positif < n yang relatif prima dengan n (Munir, 2006). Dua bilangan bulat a dan b dikatakan relatif prima jika gcd(a,b) = 1(pembagi bersama terbesar dari a dan b adalah 1). Jika $n = pq$ (p dan q bilangan prima), maka $\varphi(n) = \varphi(p).\varphi(q) = (p-1)(q-1)$. Contoh: φ(15)= φ(3) φ(5)=2x4=8 buah bilangan bulat yang relatif prima terhadap 15, yaitu 1,2,4,7,8,11,13,14.

2). Algoritma Euclidean: Algoritma ini digunakan untuk mencari nilai pembagi persekutuan terbesar (PBB) dari dua bilangan bulat (Munir, 2006). Algoritma ini didasarkan pada pernyataan bahwa ada dua buah bilangan bulat tak negatif

yakni m dan n dimana nilai $m \geq n$. Adapun tahap-tahap pada algoritma *Euclidean* adalah:

1. Jika $n = 0$ maka m adalah PBB(m, n); stop. Kalau tidak (yaitu $n \neq 0$) lanjutkan ke langkah nomor 2.
2. Bagilah m dengan n dan misalkan sisanya adalah r.
3. Ganti nilai m dengan nilai n dan nilai n dengan nilai r, lalu ulang kembali ke langkah nomor 1.

Algoritma *Euclidean* dapat digunakan untuk mencari dua buah bilangan bulat yang relatif prima. Dua buah bilangan bulat dikatakan relatif prima jika GCD dari kedua bilangan bernilai 1. Contoh: Hitung nilai GCD(100, 64) dan GCD(43, 19).

100 mod 64	43 mod 19
64 = 1 . 36 + 28	43 = 2 . 19 + 5
36 = 1 . 28 + 8	19 = 3 . 5 + 4
28 = 2 . 8 + 4	5 = 1 . 4 + 1
8 = 2 . 4 + 0	4 = 4 . 1 + 0
Nilai GCD(100, 64) = 4	Nilai GCD(43, 19) = 1
GCD(100, 64) ≠ 1	

3). Metode Fast Exponentiation: Metode ini digunakan untuk menghitung operasi pemangkatan besar bilangan bulat modulo dengan cepat (Munir, 2006). Metode ini berdasarkan pada pernyataan berikut ini:

$ab \bmod m = [(a \bmod m)(b \bmod m)] \bmod m$
$(abc..) \bmod m = [(a \bmod m)(b \bmod m)(c \bmod m)....] \bmod m$

Contoh:Hitung nilai dari $77^{40} \bmod 388$.

$77^{40} = 77^{32} . 77^{8}$
$77^{2} \bmod 388 = 5929 \bmod 388 = 109$
$77^{4} \bmod 388 = 77^{2} . 77^{2} \bmod 388 = [(77^{2} \bmod 388) . (77^{2} \bmod 388)] \bmod 388$
$= 109^{2} \bmod 388$
$= 241$
$77^{8} \bmod 388 = 77^{4} . 77^{4} \bmod 388 = [(77^{4} \bmod 388) . (77^{4} \bmod 388)] \bmod 388$
$= 241^{2} \bmod 388$
$= 269$
$77^{16} \bmod 388 = 77^{8} . 77^{8} \bmod 388 = [(77^{8} \bmod 388) . (77^{8} \bmod 388)] \bmod 388$
$= 269^{2} \bmod 388$
$= 193$
$77^{32} \bmod 388 = 77^{16} . 77^{16} \bmod 38 = [(77^{16} \bmod 388) . (77^{16} \bmod 388)] \bmod 388$
$= 193^{2} \bmod 388$
$= 1$

$77^{40} \bmod 388 = [(77^{32} \bmod 388) . (77^{8} \bmod 388)] \bmod 388 = 1 . 269 \bmod 388 = 269$
Jadi, nilai dari $77^{40} \bmod 388 = 269$.

4). Proses Pembangkitan Kunci: Dalam proses pembangkitan kunci baik kunci publik maupun kunci privat pada algoritma RSA, dapat dilakukan dengan langkah-langkah sebagai berikut.

- Pilih dua buah bilangan prima secara random yakni p dan q akan tetapi nilai $p \neq q$.

2	3	5	7	11	13
17	19	23	29	31	37
41	43	47	53	59	61
67	71	73	79	83	.....

- Contoh: Nilai p adalah 19 dan q adalah 41, p dan q adalah bilangan prima yang telah diperoleh dengan metode *the sieve of* eratosthenes.
- Hitung $n = p.q$, sehingga nilai n = 19 x 41 adalah 779
- Hitung $\varphi(n) = (p-1)(q-1)$, sehingga nilai φ(n) = (19-1)(41-1) adalah 720
- Kemudian bangkitkan kunci publik (e), dimana nilai e relatif prima terhadap φ(n). Nilai GCD(ϕ(n), e) harus bernilai 1. Untuk menentukan nilai kunci publik (e) yang relatif prima terhadap φ(n) dapat ditunjukkan pada perhitungan dibawah ini.

Mulai dari	**Nilai GCD(720, e)**
e = 2	720 mod 2 = 0 GCD(2, 720) = 2
e = 3	720 mod 3 = 0 GCD(3, 720) = 3
e = 4	720 mod 4 = 0 GCD(4, 720) = 4
e = 5	720 mod 5 = 0 GCD(5, 720) = 5
e = 6	720 mod 6 = 0 GCD(6, 720) = 6
e = 7	720 mod 7 = 6 7 mod 6 = 1 6 mod 1 = 0 GCD(7, 720) = 1

Jadi, nilai dari kunci publik (e) yang diperoleh adalah 7.

- Hitung kunci privat (d) dengan menggunakan persamaan $d = \frac{1+k.\phi(n)}{e}$. Nilai k dapat dihitung

dengan mencoba nilai-nilai= 1,2,3,4...sehingga diperoleh nilai *d* bilangan bulat.

Nilai *k*	$d = \frac{1 + k \cdot \phi(n)}{e}$ **Persamaan**	**Hasil**
1	$d = \frac{1 + 1 \cdot 720}{7}$	103
2	$d = \frac{1 + 2 \cdot 720}{7}$	205.857142857142857
3	$d = \frac{1 + 3 \cdot 720}{7}$	308.71428571428571
4	$d = \frac{1 + 4 \cdot 720}{7}$	411.57142857142857
....	.....	.....

Jadi, nilai dari kunci privat (d) yang diperoleh adalah 103. Pada kunci publik terdiri atas:

a. n, modulus yang digunakan.
b. e, kunci publik, kunci untuk enkripsi

Pada kunci privat terdiri atas:

a. n, modulus yang digunakan.
b. d, kunci privat (kunci untuk dekripsi) yang harus dijaga kerahasiaannya.

*5). Proses Enkripsi Pesan.*Dalam enkripsi pesan menggunakan algoritma RSA dapat dianalogikan seperti proses pengiriman surat yang dilakukan oleh *user* A (sebagai pengirim) dan *user* B (sebagai penerima). Maka *user* A harus melakukan beberapa langkah sebagai berikut:

- *User* A menentukan kunci publik (e) dan modulus (n) dari pesan terlebih dahulu dengan melakukan proses pembangkitan kunci sehingga mendapatkan nilai e = 7 dan nilai n = 779.
- Kemudian *user* A memasukkan plainteks yang akan dienkripsi contohnya **Teknologi Informasi USU**. Plainteks yang telah di-*input* akan diubah sesuai tabel ASCII di bawah ini, menjadi $P_1 - P_{20}$

Dari tabel ASCII diperoleh hasil sebagai berikut.

P_1 = 84	P_2 = 101	P_3 = 107	P_4 = 110
P_5 = 111	P_6 = 108	P_7 = 111	P_8 = 103
P_9 = 105	P_{10} = 32	P_{11} = 73	P_{12} = 110
P_{13} = 102	P_{14} = 111	P_{15} = 113	P_{16} = 109
P_{17} = 97	P_{18} = 115	P_{19} = 105	P_{20} = 32
P_{21} = 85	P_{22} = 83	P_{23} = 85	

- Setiap hasil yang diperoleh akan dienkripsi menjadi blok C_i dengan rumus $C_i = P_i^e \, mod \, n$. Pada tahap ini *user* B memberikan kunci publik ke pada *user* A yaitu e = 7 dan nilai n = 779. *User* A melakukan enkripsi setiap blok pesan sebagai berikut:

P_i	**Cipherteks** $C_i = P_i^e \bmod n$	P_i	**Cipherteks** $C_i = P_i^e \bmod n$
84	46	102	254
101	522	111	511
107	31	113	474
110	507	109	79
111	511	97	280
108	48	115	39
111	511	105	414
103	730	32	542
105	414	85	137
32	542	83	83
73	378	85	137
110	507		

- Jadi, hasil dari enkripsi atau cipherteks yang akan dikirimkan ke *user* b adalah 8-750-373-222-568-561-568-692-72-378-222-292-568-189-3-546-267-72-232-83-232. Setelah terenkripsi maka *user* A dapat mengirimkan chiperteks pada *user* B.

6). Proses Dekripsi Pesan. *User* B menerima cipherteks(C_i) dari *user* A. Kemudian *user* B melakukan dekripsi pesan dari *user* A yang masih berupa chiperteks. Setiap blok cipherteks(C_i) didekripsi kembali menjadi blok P_i dengan rumus $P_i = C_i^d \; mod \; n$. Dekripsi dilakukan dengan menggunakan kunci privat d = 67, kemudian blok–blok chiperteks yang sudah ada didekripsikan sebagai berikut :

C_i	**Plainteks** $P_i = C_i^d \bmod n$	C_i	**Plainteks** $P_i = C_i^d \bmod n$
8	84	292	102
750	101	568	111
373	107	189	113
222	110	3	109
568	111	546	97
561	108	267	115
568	111	72	30
692	103	232	85

72	105	83	83
378	73	232	85
222	110		

- Semua hasil cipherteks yang diperoleh pada proses dekripsi akan diubah kembali menjadi plainteks dengan mencocokkan karakter yang ada pada tabel ASCII.
- Jadi bentuk plainteks yang diperoleh adalah **Teknologi Informasi USU**.

III. ANALISA

Dalam algoritma RSA, ada tiga proses yang harus dilakukan yaitu: peroses pembangkitan kunci, proses enkripsi dan proses dekripsi.

A. *Pembangkitan Kunci*

Proses penentuan bilangan prima yang digunakan dapat dilihat pada bagian metodologi sebelumnya dengan metode *The Sieve Of Eratosthenes*. Hasil dari proses ini adalah pasangan kunci public (enkripsi) dan kunci privat (deskripsi).

*1). Enkripsi.*Proses enkripsi merupakan proses untuk mengubah plainteks menjadi cipherteks. Proses enkripsidilakukan dengan menggunakan kunci publik yang diperoleh. Algoritma proses enkripsi ditunjukkan pada Tabel I.

TABLE I. PROSES ENKRIPSI

Pesan teks = P Kunci Publik = {e, n} Enkripsi : $C_i = P_i^e$ mod n dimana C_i adalah *ciphernumber* ke i

*2). Deskripsi.*Proses dekripsi merupakan proses untuk mengubah cipherteks menjadi plainteks. Proses dekripsi dapat dilakukan dengan menggunakan kunci privat yang diperoleh. Algoritma proses dekripsi ditunjukkan pada Tabel II.

TABLE II. PROSES DEKRIPSI

Cipherteks = C Kunci Privat = {d, n} Enkripsi : $P_i = C_i^d$mod n dimana P_i adalah *plainnumber* ke i

B. *Deskripsi Perangkat Lunak.*

Pada perangkat lunak ini, pengguna harus mempunyai akun *Gmail* yang benar-benar *valid* sehingga akun tersebut dapat digunakan sebagai alamat dari si pengirim dan si penerima. Perangkat lunak dibangun dengan menggunakan bahasa program Visual Basic .NET 2008.

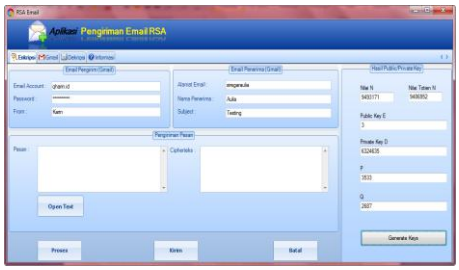
IV. IMPLEMENTASI DAN UJICOBA

Seperti yang dirancang pada tahap perancangan, tampilan perangkat lunak begitu dijalankan merupakan tampilan pembuatan kunci. Adapun beberapa batasan masalah yaitu:

1. Pesan yang dapat di-*input* dan yang akan dienkripsi hanya berupa teks dalam format .txt. Batasan ini dapat ditunjukkan pada *interface* yang hanya memuat *output* pada *textbox*. Selain itu, pada aplikasi yang dibuat sudah terdapat *namespace* yang mengambil *class* untuk mendeklarasikan teks sebagai *input* dan *output*.
2. Proses pengiriman *email* melalui akun *Gmail*. Hal ini dikarenakan perangkat lunak yang dibuat memakai settingan SMTP untuk *Gmail*.
3. Pada perangkat lunak ini, algoritma yang digunakan hanya algoritma RSA. Batasan ini dapat diperlihatkan pada saat pemanggilan *class* yang diberi nama **RSAEncryprion.vb** dimana di dalam *class* tersebut terdapat fungsi-fungsi sesuai algoritma RSA.

Pada aplikasi ini terdapat 3 tahap yang akan dilakukan oleh *user* yaitu proses enkripsi, proses penerimaan pesan dan proses dekripsi.

1. Tampilan pada **Tab Enkripsi.** Dalam **Tab Enkripsi** terdapat proses pembangkitan kunci untuk menghasilkan kunci publik dan kunci privat. Kunci dihasilkan dari bilangan prima yang diacak pada kotak yang tersedia. Untuk membangkitkan pasangan kunci publik dan privat adalah dari tombol **Generate *Keys***.Proses pembangkitan kunci dapat ditunjukkan pada Gambar 1.



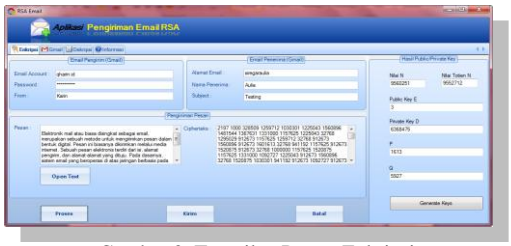
Gambar 1. Tampilan Pembangkitan Kunci Pada Tab Enkripsi

Setelah tahap inisialisasi data yang sesuai dengan akun *Gmail* dan pembangkitan kunci, *user* memasukkan plainteks yang akan dikirim pada *textbox* pesan yang tersedia dengan memilih tombol **Open Text**. Saat tombol ini diklik, maka akan muncul tampilan pada Gambar 2.

Gambar 2. Tampilan Buka *File*Teks

File yang dipilih akan masuk ke dalam *textbox* pesan. Setelah pesan yang diinginkan muncul, maka pesan tersebut harus dienkripsi dengan memilih tombol **Proses**. Pada saat proses enkripsi selesai akan muncul cipherteks yang menjadi pesan untuk dikirim

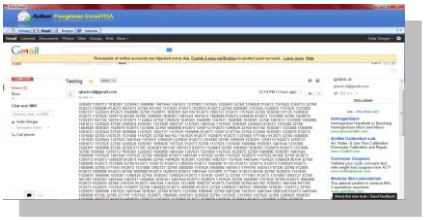
pada *user* lain. Tampilan proses enkrispi dapat ditunjukkan pada Gambar 3.



Gambar 3. Tampilan Proses Enkripsi

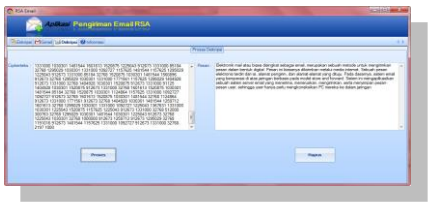
Plainteks yang telah diubah menjadi cipherteks kemudian dikirim ke alamat *email* yang dituju.

2. Tampilan pada **Tab *Gmail.*** Pada **Tab *Gmail*** akan muncul halaman *web* dari *Gmail* sehingga pengguna dapat langsung mengakses akun *Gmail*. Pesan yang terkirim akan langsung masuk pada akun *Gmail* penerima Tampilan tab *Gmail* dapat dilihat pada Gambar 4.



Gambar 4. Tampilan Pada Kotak Masuk *Gmail*

3. Tampilan pada **Tab Dekripsi. Tab Dekripsi** merupakan tempat untuk melakukan proses dekripsi dari pesan yang terkirim ke akun *Gmail*. Pesan yang berupa cipherteks tersebut di-*copy* dan di-*paste* ke dalam *textbox* cipherteks yang tersedia. Tampilan proses dekripsi pada tab dekripsi dapat ditunjukkan pada Gambar 5.



Gambar 5. Tampilan Proses Dekripsi

V. KESIMPULAN

Dapat diambil kesimpulan bahwa perangkat lunak yang dibuat sudah mampu memberikan keamanan pada saat pengiriman *email*. Kunci privat yang dibangkitkan selalu berubah-ubah nilainya walau terkadang nilai kunci publik yang dihasilkan nilainya sama. Pembangkitan bilangan acak diimplementasikan dalam suatu fungsi *Math.random*. Sementara pada proses penentuan bilangan prima digunakan fungsi khusus *Math.sqrt* yang sesuai dengan metode *The Sieve of Eratosthenes* untuk menentukan nilai akar kuadrat dari suatu bilangan sehingga diharapkan pengecekan bilangan prima bisa dilakukan lebih cepat dan menghemat memori. Pembuatan aplikasi simulasi kriptografi berbasis algoritma RSA ini dapat membuktikan bahwa algoritma kriptografi RSA dapat diimplementasikan dengan menggunakan fungsi-fungsi khusus yang tersedia dalam bahasa pemrograman Visual Basic .NET 2008 walaupun masih bersifat *stand alone* dan hanya menampilkan proses dari algoritma RSA saja.

DAFTAR PUSTAKA

[1] Alghazali, M. R. 2010. *Sieve of Eratosthenes, Algoritma Bilangan Prima*. Makalah. Bandung: Institut Teknologi Bandung.

[2] Ananda *et al*. 2009. *Algoritma dan Pemrograman*. Makalah. Bandung: Politeknik Telkom.

[3] Ariyus, D. 2008. *Pengantar Ilmu Kriptografi (Teori, Analisis dan Implementasi)*. Edisi ke-1. Yogyakarta: Andi.

[4] Brian, F. 2000. RSA Releases Patent Early. *InfoWord*. 22: hal. 27.

[5] Cazelais, G. 16 Juli 2010. *Sieve of Eratosthenes*.

[6] Firasyan, T. 2011. *Penggunaan Algoritma RSA untuk Keamanan Transaksi Online Berbasis Aplikasi Mobil*e. Tugas Akhir. Surabaya: Institut Teknologi Sepuluh November Surabaya.

[7] Fithria, N. 2007. *Jenis–Jenis Serangan Terhadap Kriptografi*. Makalah. Bandung: Institut Teknologi Bandung.

[8] Halvorson, M. 2008. *Microsoft Visual Basic 2008 Step by Step*. Microsoft Press.

[9] Karls, M.A. 2010. Codes, Cipher, and Cryptography-An Honors Colloquium. *Primus*. 20: hal. 21-22.

[10] Kashogi, A. 2007. *Menentukan primalitas semua bilangan yang terdapat pada selang tertentu secara brute force*. Makalah. Bandung: Institut Teknologi Bnadung.

[11] Munandar, D. Y. 2007. *Aplikasi Pengamanan Pesan Pada Mail Client Menggunakan Algoritma RC6*. Tugas Akhir. Jakarta: Universitas Komputer Indonesia.

[12] Munir, R. 2006. *Kriptografi*. Edisi ke-1. Bandung: Informatika.

[13] Mollin, R. A. 2007. *An Introduction to Cryptography*. 2nd edition. New York: Taylor & Francis Group.

[14] Permana *et al*. 2008. *Teori Email dan Forum Komunikasi di dalam Email*. Makalah. Yogyakarta: Universitas Pembangunan Nasional "VETERAN".

[15] Pressman, R.S. 2002. *Rekayasa Perangkat Lunak (Pendekatan Satu)*. Terjemahan LN Harmaningrum. Yogyakarta: Andi.

[16] Putra, R.A. 2007. *Aplikasi Kriptografi untuk Proteksi dan Keamanan Sistem Informasi*. Makalah. Bandung: Institut Teknologi Bandung.

[17] Sulistyanto, H. 2004. Autentikasi dalam Basis Data Jaringan Menggunakan Kriprosistem Kunci Publik RSA. *Jurnal Teknik Elektro dan Komputer Emitor*. **Volume 4**: hal. 40-41.

[18] Surahman, A. 2009. *Membuat dan Berkomunikasi dengan Menggunakan Email*. Makalah. Yogyakarta: Universitas Gajah Mada.

[19] Umniati, N. 2002. *Perbandingan Algoritma dalam Enkripsi antara Conventional Cryptosystems dan Public Key Cryptosystems*. Artikel. Jakarta: Universitas Gunadarma.